

AMENDMENTS TO THE CLAIMS

1. (Previously Presented) In an object persistence management system, a many-to-many relationship manager comprising:

a plurality of related objects;

a junction table storing relationships between said related objects; and,

a plurality of corresponding decentralized links, each said link corresponding to one of said objects, each said link persisting state information for said corresponding object in an associated object table, and managing said junction table responsive to changing relationships with others of said related objects.

2. (Original) The many-to-many relationship manager of claim 1, further comprising a counter-operation management protocol performed in said corresponding links for removing conflicted state information in said corresponding links without persisting said conflicted state information in said junction table.

3. (Original) The many-to-many relationship manager of claim 2, wherein each of said corresponding links comprises a state management operations buffer, said buffer storing directives for adding selected key-pair entries to and removing selected key-pair entries from said junction table.

4. (Original) The many-to-many relationship manager of claim 3, wherein said counter-operation management protocol comprises an interface through which operations in said buffer

and corresponding counter-operations in associated buffers of related links can be identified and removed, each said counter-operation specifying a junction table management operation for a particular key-pair entry in said associated buffer which is opposite to an operation in said buffer which specifies a junction table management operation also for said particular key-pair entry.

5. (Previously Presented) A method of managing a many-to-many relationship in an object persistence management system comprising the steps of:

- detecting a relationship change with a related object;
- storing a directive in a buffer, said directive specifying a management operation for changing said relationship in a junction table; and,
- searching for an opposite directive in a buffer associated with said related object; and,
- performing said stored directive only if the opposite directive has not been stored in said buffer associated with said related object.

6. (Original) The method of claim 5, wherein said storing step comprises the step of storing a directive in said buffer which specifies one of adding or removing a key-pair entry in said junction table.

7. (Original) The method of claim 6, wherein said performing step comprises the steps of:

- performing said specified adding or removing of said key-pair entry only if a corresponding opposite directive specifying a respective removing or adding of said key-pair entry is not detected in said buffer of said associated object; and,

Application No.: 10/026,387

responsive to detecting said corresponding opposite directive, removing both said directive and opposite directive from both said buffers.

Claims 8-11 (Cancelled)